

EXHIBIT A

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION**

Implicit, LLC,

Plaintiff,

v.

Trend Micro, Inc.,

Defendant.

Case No. 6:16-cv-80-JRG
LEAD CASE

CLAIM CONSTRUCTION MEMORANDUM OPINION AND ORDER

Before the Court is the opening claim construction brief of Implicit, LLC (“Plaintiff”) (Dkt. No. 101, filed on January 17, 2017),¹ the response of Trend Micro, Inc., Ericsson Inc., and Huawei Technologies USA, Inc. (collectively “Defendants”) (Dkt. No. 103, filed on January 31, 2017), and the reply of Plaintiff (Dkt. No. 106, filed on February 10, 2017). The Court held a hearing on the issues of claim construction and claim definiteness on February 28, 2017. Having considered the arguments and evidence presented by the parties at the hearing and in their briefing, the Court issues this Order.

¹ Citations to the parties’ filings are to the filing’s number in the docket (Dkt. No.) and pin cites are to the page numbers assigned through ECF.

Table of Contents

I.	BACKGROUND	3
A.	The Demultiplexing Patents.....	4
A-1.	Technology	4
A-2.	Related Litigation.....	5
B.	The Applet Patents	6
II.	LEGAL PRINCIPLES	9
A.	Claim Construction	9
B.	Departing from the Ordinary Meaning of a Claim Term.....	11
III.	AGREED CONSTRUCTIONS.....	12
IV.	CONSTRUCTION OF DISPUTED TERMS.....	13
A.	The Demultiplexing Patents.....	13
A-1.	“sequence of routines” and “sequence of two or more routines”	13
A-2.	“processing packets” and “process ... packets”.....	20
A-3.	“create”	23
A-4.	“the packet of the message”	24
A-5.	“list of conversion routines”	28
B.	The Applet Patents	30
B-1.	“form of the application”	30
B-2.	“resource”	34
B-3.	“generating the identified form of the application from another form of the application” and “generated the identified form of the application from another form of the application”	37
B-4.	“source code that is in a form based on the specified one or more client parameters for the first client computer”	40
B-5.	“a specific form of the particular applet that includes source code, based on the specified one or more parameters in the applet request, wherein the specific form complies with the specified one or more parameters”.....	42
B-6.	“transformation operation”	43
V.	CONCLUSION	45

I. BACKGROUND

Plaintiff alleges infringement of five U.S. Patents: No. 6,324,685 (the “‘685 Patent”), No. 8,694,683 (the “‘683 Patent”), No. 8,856,779 (the “‘779 Patent”), No. 9,270,790 (the “‘790 Patent”), and No. 9,325,740 (the “‘740 Patent”) (collectively, the “Asserted Patents”). The ‘685 Patent is entitled “Applet Server That Provides Applets in Various Forms.” The application leading to the ‘685 Patent was filed on March 18, 1998 and the patent issued on November 27, 2001. The ‘683 Patent is entitled “Method and System for Data Demultiplexing.” The application leading to the ‘683 Patent was filed on June 6, 2013 and the patent issued on April 8, 2014. The ‘779 Patent is entitled “Application Server for Delivering Applets to Client Computing Devices in a Distributed Environment.” The application leading to the ‘779 Patent was filed on October 10, 2011 and the patent issued on October 7, 2014. The ‘790 Patent is entitled “Method and System for Data Demultiplexing.” The application leading to the ‘790 Patent was filed on March 31, 2014 and the patent issued on February 23, 2016. The ‘740 Patent is entitled “Application Server for Delivering Applets to Client Computing Devices in a Distributed Environment.” The application leading to the ‘740 Patent was filed on October 6, 2014 and the patent issued on April 26, 2016.

The Asserted Patents are part of two patent families: the Demultiplexing Patents (the ‘683 Patent and the ‘790 Patent) and the Applet Patents (the ‘685 Patent, the ‘779 Patent, and the ‘740 Patent). With respect to the Demultiplexing Patents: The ‘790 Patent claims priority to the ‘683 Patent’s application as a continuation. Through a series of continuation applications, the ‘790 Patent and the ‘683 Patent each claim priority to an application filed on December 29, 1999 and issued as U.S. Patent No. 6,629,163 (the “‘163 Patent”). With Respect to the Applet Patents: The ‘740 Patent claims priority to the ‘779 Patent’s application as a continuation. The ‘740 Patent and

the '779 Patent each claim priority to the application that issued as the '685 Patent, through a series of continuation applications.

A. The Demultiplexing Patents

A-1. Technology

The Demultiplexing Patents are generally directed to technology for computer message-exchange processing and more specifically to technology for dynamically converting the form of the messages as the messages are being exchanged.

The abstract of the '683 Patent provides:

A method and system for demultiplexing packets of a message is provided. The demultiplexing system receives packets of a message, identifies a sequence of message handlers for processing the message, identifies state information associated with the message for each message handler, and invokes the message handlers passing the message and the associated state information. The system identifies the message handlers based on the initial data type of the message and a target data type. The identified message handlers effect the conversion of the data to the target data type through various intermediate data types.

The abstract of the '790 Patent provides:

A method and system for demultiplexing packets of a message is provided. The demultiplexing system receives packets of a message, identifies a sequence of message handlers for processing the message, identifies state information associated with the message for each message handler, and invokes the message handlers passing the message and the associated state information. The system identifies the message handlers based on the initial data type of the message and a target data type. The identified message handlers effect the conversion of the data to the target data type through various intermediate data types.

Claim 1 of the '683 Patent, provided here as an example, recites:

1. A first apparatus for receiving data from a second apparatus, the first apparatus comprising:
 - a processing unit; and
 - a memory storing instructions executable by the processing unit to:
 - create, based on an identification of information in a received packet of a message, a path that includes one or more data structures that indicate a sequence of routines for processing packets in the message;
 - store the created path; and

process subsequent packets in the message using the sequence of routines indicated in the stored path, wherein the sequence includes a routine that is used to execute a Transmission Control Protocol (TCP) to convert one or more packets having a TCP format into a different format.

Claim 8 of the '790 Patent, provided here as an example, recites:

8. An apparatus, comprising:
a processing unit; and
a memory storing instructions executable by the processing unit to:
receive one or more packets of a message;
identify, using an IP address and one or more port addresses located in one of the received packets, a sequence of two or more routines for processing packets in the message; and
process the one or more received packets using the identified sequence of routines, wherein the sequence includes a routine that is executable to perform a Transmission Control Protocol (TCP) to convert at least one of the packets of the message into a different format.

A-2. Related Litigation

Two of the Demultiplexing Patents have previously been litigated in the U.S. District Court for the Northern District of California. That court construed the '163 Patent in *Implicit Networks, Inc. v. F5 Networks, Inc.*, No. 3:10-cv-3365-SI, 2012 U.S. Dist. LEXIS 27238 (N.D. Cal. Feb. 29, 2012) ("F5 Networks I"). The California court later construed the '683 Patent in *Implicit L.L.C. v. F5 Networks, Inc.*, No. 3:14-cv-2856-SI, 2015 U.S. Dist. LEXIS 60197 (N.D. Cal. May 6, 2015) ("F5 Networks II"). The F5 Networks I and F5 Networks II constructions relate to the "sequence of routines," "sequence of two or more routines" and "list of conversion routines" limitations of the Asserted Patents.

In F5 Networks I, the court construed the term "non-predefined sequence of components" found in claims of the '163 Patent. First, the court held that the term "components" was defined in the '163 Patent to mean "software routines." 2012 U.S. Dist. LEXIS 27238, at *9–10. Then the court determined that a description of the prior art found in the '163 Patent and patent-owner statements made during reexamination of the '163 Patent amounted to disclaimer of *preconfigured*

sequences of software routines. *Id.* at *10–13. Thus, the court construed “non-predefined sequence of components” as “a sequence of software routines that was not identified before the first packet of a message was received.” *Id.* at *13.

In *F5 Networks II*, the court construed the terms “sequence of routines” and “list of conversion routines” found in claims of the ’683 Patent. These terms are presently before the Court, as is the similar term “sequence of two or more routines” from the ’790 Patent. The *F5 Networks II* court determined that the disclaimer of *preconfigured* sequences of software routines was tied to the invention described in the ’163 Patent—and not limited to specific language recited in the claims of the ’163 Patent. 2015 U.S. Dist. LEXIS 60197, at *9–12. *F5 Networks II* reiterated its analysis of the description of the prior art in the ’163 and ’683 Patents and the patent owner’s explanation of that disavowal in the reexamination of the ’163 Patent. *Id.* at *34–37 (noting that the patent owner “devoted an entire section of its [reexamination] response to further explain these prior art disavowals in the first column of the specification shared by the ’163 and ’683 patents”). Ultimately, *F5 Networks II* held that the patent owner “made it definitively clear to the PTO and the public that the sequence of routines (or ‘path’) as disclosed in the ’163 patent specification is not configured before receiving the first packet of a message” and that this disclaimer applies equally to the ’683 Patent. *Id.* at *37–39. The court construed “sequence of routines” and “list of conversion routines” as “a sequence of software routines that was not identified (i.e., configured) prior to receiving a first packet of the message” and “a list of software routines that was not identified (i.e., configured) prior to receiving a first packet of the message,” respectively. *Id.* at *42.

B. The Applet Patents

In general, the Applet Patents are directed to server technology for providing applets and applications to a client computer.

The abstract of the '685 Patent provides:

The present invention is an applet server which accepts requests for applets from client computers. A request specifies the format in which an applet is to be delivered to the requesting client computer. The applet server has a cache which it uses to store applets for distribution to client computers. If the specified form of the requested applet is available in the cache, the applet server transmits the applet to the requesting client. If the applet is not available in the cache, the server will attempt to build the applet from local resources (program code modules and compilers) and transformer programs (verifiers and optimizers). If the applet server is able to build the requested applet, it will then transmit the applet to the requesting client computer. If the applet server is unable to build the requested applet, it will pass the request to another applet server on the network for fulfillment of the request.

The abstract of the '779 Patent provides:

An applet server accepts requests for applets from client computers. A request specifies the format in which an applet is to be delivered to the requesting client computer. The applet server has a cache used to store applets for distribution to client computers. If the specified form of the requested applet is available in the cache, the applet server transmits the applet to the requesting client. If the applet is not available in the cache, the server will attempt to build the applet from local resources (program code modules and compilers) and transformer programs (verifiers and optimizers). If the applet server is able to build the requested applet, it will transmit the applet to the requesting client computer. If the applet server is unable to build the requested applet, it will pass the request to another applet server on the network for fulfillment of the request.

The abstract of the '740 Patent provides:

An applet server accepts requests for applets from client computers. A request specifies the format in which an applet is to be delivered to the requesting client computer. The applet server has a cache used to store applets for distribution to client computers. If the specified form of the requested applet is available in the cache, the applet server transmits the applet to the requesting client. If the applet is not available in the cache, the server will attempt to build the applet from local resources (program code modules and compilers) and transformer programs (verifiers and optimizers). If the applet server is able to build the requested applet, it will transmit the applet to the requesting client computer. If the applet server is unable to build the requested applet, it will pass the request to another applet server on the network for fulfillment of the request.

Claim 1 of the '685 Patent, provided here as an example, recites:

1. A method in a server computer for providing applications to client computers, the method comprising:

receiving a request from a client computer, the request identifying an application and identifying a form of the application, the identified form being one of a plurality of available forms;
in response to receiving the request,

generating the identified form of the application from another form of the application; and

sending the identified form of the application to the client computer; and
caching the identified form of the application so that when another request is received for the application in the identified form, the identified form of the application can be sent without regenerating the identified form of the application.

Claim 12 of the '779 Patent, provided here as an example, recites:

12. A non-transitory computer-readable storage medium having stored thereon instructions that are executable to cause a computer system to perform operations comprising:

receiving an applet request from a first client computer for a particular applet, wherein the applet request specifies one or more parameters for the particular applet that are based on one or more characteristics of the client computer;
acquiring a specific form of the particular applet that includes source code, based on the specified one or more parameters in the applet request, wherein the specific form complies with the specified one or more parameters; and
sending the specific form of the particular applet to the first client computer in response to the applet request.

Claim 1 of the '740 Patent, provided here as an example, recites:

1. A non-transitory computer-readable storage medium having stored thereon instructions that are executable to cause a computer system to perform operations comprising:

receiving, at the computer system, a first HTTP request from a first client computer for a resource, wherein the resource includes source code;

producing, by the computer system, the resource for the first client computer, wherein the producing includes:

conveying, by the computer system, a request for the resource to an external network;

receiving, at the computer system, the resource from the external network; and

performing, by the computer system, a transformation operation on the resource; and

sending, by the computer system, the produced resource to the first client computer in response to the first HTTP request.

II. LEGAL PRINCIPLES

A. Claim Construction

“It is a ‘bedrock principle’ of patent law that ‘the claims of a patent define the invention to which the patentee is entitled the right to exclude.’” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (quoting *Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). To determine the meaning of the claims, courts start by considering the intrinsic evidence. *Id.* at 1313. The intrinsic evidence includes the claims themselves, the specification, and the prosecution history. *Id.* at 1314. The general rule—subject to certain specific exceptions discussed *infra*—is that each claim term is construed according to its ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the patent. *Id.* at 1312–13; *see also Azure Networks, LLC v. CSR PLC*, 771 F.3d 1336, 1347 (Fed. Cir. 2014) (“There is a heavy presumption that claim terms carry their accustomed meaning in the relevant community at the relevant time.”) (vacated on other grounds).

“[I]n all aspects of claim construction, ‘the name of the game is the claim.’” *Apple Inc. v. Motorola, Inc.*, 757 F.3d 1286, 1298 (Fed. Cir. 2014) (quoting *In re Hiniker Co.*, 150 F.3d 1362, 1369 (Fed. Cir. 1998)). First, a term’s context in the asserted claim can be instructive. *Phillips*, 415 F.3d at 1314. Other asserted or unasserted claims can also aid in determining the claim’s meaning, because claim terms are typically used consistently throughout the patent. *Id.* Differences among the claim terms can also assist in understanding a term’s meaning. *Id.* For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation. *Id.* at 1314–15.

“[C]laims [also] ‘must be read in view of the specification, of which they are a part.’” *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc)).

“[T]he specification ‘is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.’” *Id.* (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)). However, “[a]lthough the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims.”” *Comark Commc’ns, Inc. v. Harris Corp.*, 156 F.3d 1182, 1187 (Fed. Cir. 1998) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988)); *see also Phillips*, 415 F.3d at 1323. “[I]t is improper to read limitations from a preferred embodiment described in the specification—even if it is the only embodiment—into the claims absent a clear indication in the intrinsic record that the patentee intended the claims to be so limited.” *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 913 (Fed. Cir. 2004).

The prosecution history is another tool to supply the proper context for claim construction because, like the specification, the prosecution history provides evidence of how the U.S. Patent and Trademark Office (“PTO”) and the inventor understood the patent. *Phillips*, 415 F.3d at 1317. However, “because the prosecution history represents an ongoing negotiation between the PTO and the applicant, rather than the final product of that negotiation, it often lacks the clarity of the specification and thus is less useful for claim construction purposes.” *Id.* at 1318; *see also Athletic Alternatives, Inc. v. Prince Mfg.*, 73 F.3d 1573, 1580 (Fed. Cir. 1996) (ambiguous prosecution history may be “unhelpful as an interpretive resource”).

Although extrinsic evidence can also be useful, it is “less significant than the intrinsic record in determining the legally operative meaning of claim language.”” *Phillips*, 415 F.3d at 1317 (quoting *C.R. Bard, Inc.*, 388 F.3d at 862). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might

use claim terms, but they may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id.* at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert's conclusory, unsupported assertions as to a term's definition are not helpful to a court. *Id.* Thus, extrinsic evidence is typically "less reliable than the patent and its prosecution history in determining how to read claim terms." *Id.* The Supreme Court recently explained the role of extrinsic evidence in claim construction:

In some cases, however, the district court will need to look beyond the patent's intrinsic evidence and to consult extrinsic evidence in order to understand, for example, the background science or the meaning of a term in the relevant art during the relevant time period. *See, e.g., Seymour v. Osborne*, 11 Wall. 516, 546 (1871) (a patent may be "so interspersed with technical terms and terms of art that the testimony of scientific witnesses is indispensable to a correct understanding of its meaning"). In cases where those subsidiary facts are in dispute, courts will need to make subsidiary factual findings about that extrinsic evidence. These are the "evidentiary underpinnings" of claim construction that we discussed in *Markman*, and this subsidiary factfinding must be reviewed for clear error on appeal.

Teva Pharm. USA, Inc. v. Sandoz, Inc., 135 S. Ct. 831, 841 (2015).

B. Departing from the Ordinary Meaning of a Claim Term

There are "only two exceptions to [the] general rule" that claim terms are construed according to their plain and ordinary meaning: "1) when a patentee sets out a definition and acts as his own lexicographer, or 2) when the patentee disavows the full scope of the claim term either in the specification or during prosecution."² *Golden Bridge Tech., Inc. v. Apple Inc.*, 758 F.3d 1362, 1365 (Fed. Cir. 2014) (quoting *Thorner v. Sony Computer Entm't Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012)); *see also GE Lighting Solutions, LLC v. AgiLight, Inc.*, 750 F.3d 1304, 1309

² Some cases have characterized other principles of claim construction as "exceptions" to the general rule, such as the statutory requirement that a means-plus-function term is construed to cover the corresponding structure disclosed in the specification. *See, e.g., CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1367 (Fed. Cir. 2002).

(Fed. Cir. 2014) (“[T]he specification and prosecution history only compel departure from the plain meaning in two instances: lexicography and disavowal.”). The standards for finding lexicography or disavowal are “exacting.” *GE Lighting Solutions*, 750 F.3d at 1309.

To act as his own lexicographer, the patentee must “clearly set forth a definition of the disputed claim term,” and “clearly express an intent to define the term.” *Id.* (quoting *Thorner*, 669 F.3d at 1365); *see also Renishaw*, 158 F.3d at 1249. The patentee’s lexicography must appear “with reasonable clarity, deliberateness, and precision.” *Renishaw*, 158 F.3d at 1249.

To disavow or disclaim the full scope of a claim term, the patentee’s statements in the specification or prosecution history must amount to a “clear and unmistakable” surrender. *Cordis Corp. v. Boston Sci. Corp.*, 561 F.3d 1319, 1329 (Fed. Cir. 2009). “Where an applicant’s statements are amenable to multiple reasonable interpretations, they cannot be deemed clear and unmistakable.” *3M Innovative Props. Co. v. Tredegar Corp.*, 725 F.3d 1315, 1326 (Fed. Cir. 2013).

III. AGREED CONSTRUCTIONS

The parties have agreed to the following constructions set forth in their Joint Claim Construction Chart Statement (Dkt. No. 107).

Term³	Agreed Construction
“message”	a collection of data that is related in some way, such as a stream of video or audio data or an email message
• ’683 Patent Claims 1, 10, 24 • ’790 Patent Claims 8, 15	
“applet”	program instructions provided as a self-contained program or as a code fragment associated with a larger application
• ’685 Patent Claims 18, 49 • ’779 Patent Claims 1, 12, 18	

³ For all term charts in this order, the claims in which the term is found are listed with the term but: (1) only the highest-level claim in each dependency chain is listed, and (2) only asserted claims identified in the parties’ Joint Claim Construction Chart Statement (Dkt. No. 107) are listed.

Term³	Agreed Construction
“application” <ul style="list-style-type: none">• ’685 Patent Claims 1, 3, 15, 18, 34, 49• ’740 Patent Claim 11	program designed to assist in the performance of a specific task
“cache” <ul style="list-style-type: none">• ’685 Patent Claims 15, 34• ’779 Patent Claims 4, 5, 6	temporary memory for storing an application/applet or a portion thereof
“caching” <ul style="list-style-type: none">• ’685 Patent Claims 1, 15• ’779 Patent Claim 19	temporarily storing in memory an application/applet or a portion thereof
“source code” <ul style="list-style-type: none">• ’779 Patent Claim 1, 12, 18• ’740 Patent Claims 1, 19	code in the form of a higher level language such as C, C++, Java, Visual Basic, ActiveX, Fortran, and Modula
“optimizing” <ul style="list-style-type: none">• ’779 Patent Claim 20• ’740 Patent Claim 15	making improvements to applets by substituting functionally equivalent code

Having reviewed the intrinsic and extrinsic evidence of record, the Court agrees with and hereby adopts the parties’ agreed constructions.

IV. CONSTRUCTION OF DISPUTED TERMS

A. The Demultiplexing Patents

A-1. “sequence of routines” and “sequence of two or more routines”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“sequence of routines” <ul style="list-style-type: none">• ’683 Patent Claims 1, 24	an ordered arrangement of software routines	an ordered arrangement of software routines that was not identified (i.e., configured) prior to receiving a first packet of the message
“sequence of two or more routines” <ul style="list-style-type: none">• ’790 Patent Claim 8	an ordered arrangement of two or more software routines	an ordered arrangement of two or more software routines that was not identified (i.e., configured) prior to receiving a first packet of the message

Because the parties' arguments and proposed constructions with respect to these terms are related, the Court addresses the terms together.

The Parties' Positions

Plaintiff submits the claim constructions in *Implicit Networks, Inc. v. F5 Networks, Inc.*, No. 3:10-cv-3365-SI, 2012 U.S. Dist. LEXIS 27238 (N.D. Cal. Feb. 29, 2012) ("F5 Networks I") and *Implicit L.L.C. v. F5 Networks, Inc.*, No. 3:14-cv-2856-SI, 2015 U.S. Dist. LEXIS 60197 (N.D. Cal. May 6, 2015) ("F5 Networks II") are not binding on Plaintiff or the Court. Dkt. No. 101 at 10–11. Plaintiff argues that the plain and ordinary meanings of these terms govern and that there was no disclaimer of preconfigured sequences. *Id.* at 11–13. According to Plaintiff, *F5 Networks II* erred in holding disclaimer and the basis of this error was conflating "sequence" with "path." *Id.* at 11–13. Plaintiff argues that, in the Demultiplexing Patents, a "path is a sequence of sessions, each session having associated with it its own sequence of conversion routines." *Id.* at 11. Plaintiff further argues that the patents explain that while sessions are dynamically configured (i.e., they are not preconfigured), the sequences of the dynamically generated sessions may be preconfigured. *Id.* at 11–12.

In addition to the claims themselves, Plaintiff cites the following intrinsic and extrinsic evidence to support its position: **Intrinsic evidence:** '683 Patent figs.5, 15, col.2 ll.44–49, col.3 ll.9–12, col.3 ll.34–35, col.3 ll.43–53, col.3 ll.62–67, col.6 ll.24–29, col.10 ll.46–49, col.11 ll.1–3, col.11 ll.19–21. **Extrinsic evidence:** *Authoritative Dictionary of IEEE Standards Terms*, "routine" and "sequence," (6th ed. 1996); *Microsoft Press Computer Dictionary*, "routine" (1997).

Defendants respond that: (1) the statements made in reexamination of the '163 Patent constitute disclaimer of preconfigured sequences of routines in the Demultiplexing Patents, and (2) Plaintiff is collaterally estopped from challenging the *F5 Networks I* holding of this disclaimer

with respect to the '163 Patent and the *F5 Networks II* holding of this disclaimer with respect to the Demultiplexing Patents. Dkt. No. 103 at 6–17. Defendants contend the patentee equated “sequence of conversion routines” and “path” in the reexamination of the '163 Patent. *Id.* at 15–16.

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: '163 Patent File Wrapper September 1, 2009 Amendment and Response to Office Action in Ex Parte Reexamination 90/010,356 (Defendants' Ex. 1, Dkt. No. 103-1), February 8, 2010 Amendment and Response to Advisory Action in Ex Parte Reexamination 90/010,356 (Defendants' Ex. 2, Dkt. No. 103-2), October 23, 2009 Interview Summary in Ex Parte Reexamination 90/010,356 (Defendants' Ex. 4, Dkt. No. 103-4).

Plaintiff replies that neither *F5 Networks I* nor *F5 Networks II* triggers collateral estoppel. Dkt. No. 106 at 5–8. With respect to *F5 Networks I*, Plaintiff contends: (1) the litigation involved only the '163 Patent, not any of the Asserted Patents and (2) the claim language there at issue—“**dynamically** identifying a **non-predefined** sequence of components”—is meaningfully different from the claim language here, thus the issues are not identical. *Id.* at 5, 7 (emphasis by Plaintiff). With respect to *F5 Networks II*, Plaintiff contends: (1) the litigation ended by settlement rather than judgment, and (2) the '790 Patent was not at issue. *Id.* at 5–8.

Plaintiff further replies that rather than disclaiming preconfigured sequences, the patentee expressly claimed such in Claim 9 of the '683 Patent, which recites a sequence that is stored “prior to receiving the packet of the message.” *Id.* at 7.

Analysis

The issue here distills to whether the inventions of the '683 and '790 Patents include identifying or creating the message-processing sequence of routines before receipt of the first packet of the message. They do not.

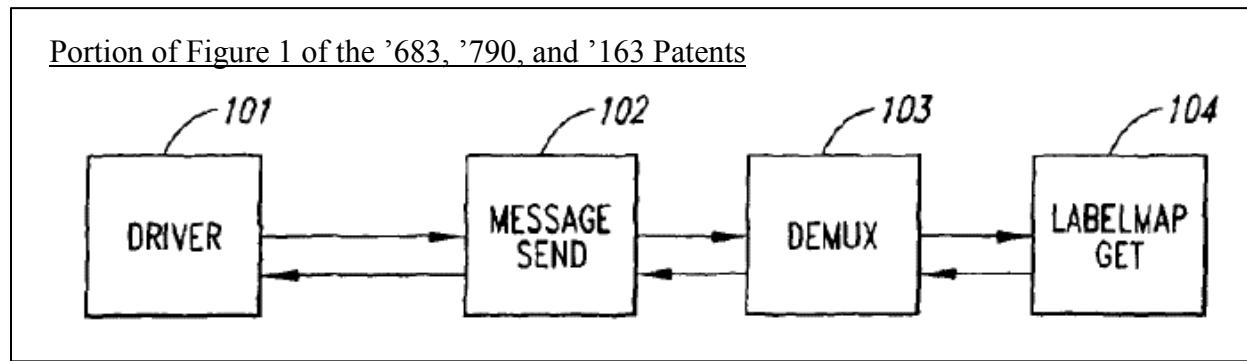
The '683 and '790 Patents are directed to dynamic configuration of conversion-processing routines used to process packetized messages into the appropriate forms. For example, the patents teach that “[i]t would be desirable to have a technique for dynamically identifying a series of conversion routines for processing data.” '683 Patent col.2 ll.4–6; '790 Patent col.2 ll.5–7 (same). The dynamic nature of the inventions is an answer to the static-nature failings of the disparaged prior art. '683 Patent col.1 l.45 – col.2 l.3; '790 Patent col.1 l.46 – col.2 l.4. Specifically, the prior-art “computer systems typically use predefined configuration information to load the correct combination of conversion routines for processing data.” '683 Patent col.1 ll.48–50; '790 Patent col.1 ll.49–51. These systems “may create a separate process for each conversion that needs to take place.” '683 Patent col.1 ll.52–54; '790 Patent col.1 ll.53–55. However, data to be converted “may take many different formats that may not be known until the data is received.” '683 Patent col.1 ll.54–57; '790 Patent col.1 ll.55–58. Because of this, “[t]he overhead of statically providing each possible series of conversion routines is very high.” '683 Patent col.1 ll.57–59; '790 Patent col.1 ll.58–60. The patentee explained this disclosure—as identically found in the '163 Patent at column 1, line 38 to column 2, line 5: it “clearly states that the invention requires the sequence of conversion routines (that form the paths) to be identified at run-time, and disavows prior art systems (like Mosberger) that use pre-configured paths, which are defined at ‘build-time’ before the first packet of a message is received.” '163 Patent File Wrapper September 1, 2009

Amendment and Response to Office Action in Ex Parte Reexamination 90/010,356 at 18 (Dkt. No. 103-1 at 18).

The patents and related prosecution history help explain the inventive dynamic-configuration approach. For example, the patents teach that “[w]hen a packet of a message is received, the conversion system in one embodiment searches for and identifies a sequence of conversion routines (or more generally message handlers) for processing the packets of the message by comparing the input and output formats of the conversion routines.” ’683 Patent col.2 ll.44–49; ’790 Patent col.2 ll.45–50. The patentee explained this passage—as identically found in the ’163 Patent at column 2 lines 40–45—to mean “the sequence of conversion routines (or ‘path’) is not configured prior to receiving the first packet of a message.” ’163 Patent File Wrapper September 1, 2009 Amendment and Response to Office Action in Ex Parte Reexamination 90/010,356 at 18 (Dkt. No. 103-1 at 18). The patents also describe the invention with reference to Figure 1 (reproduced in part below): a driver (101) sends the first packet of a message to a “message send routine” (102) which invokes a “demux routine” (103) which invokes a “labelmap get routine” (104) to identify a sequence of routines to process the message and the “demux routine” (103) subsequently enter path entries for each of the identified routines. ’683 Patent col.4 ll.1–21; ’790 Patent col.4 ll.1–21. The patentee explained that this description—as identically found in the ’163 Patent at column 3 line 65 to column 4 line 18—teaches that:

the path data structure containing the sequence of conversion routines (i.e., “sequence of components”) needed to process the message is not configured until after the first packet of the message is received. In other words, all path configuration and path creation happens at runtime, after the first packet of a message is received.

'163 Patent File Wrapper September 1, 2009 Amendment and Response to Office Action in Ex Parte Reexamination 90/010,356 at 18–19 (Dkt. No. 103-1 at 18–19).



Ultimately, the teachings of the patents regarding the inventions' dynamic-configuration approach, and disparagement of the prior-art static-configuration approach, explain that the sequence of conversion routines that together process the packets "cannot be configured before the first packet of the message is received." *See '163 Patent File Wrapper September 1, 2009 Amendment and Response to Office Action in Ex Parte Reexamination 90/010,356 at 20* (Dkt. No. 103-1 at 20); *see also, '163 Patent File Wrapper October 23, 2009 Interview Summary in Ex Parte Reexamination 90/010,356 at 10* (Dkt. No. 103-4 at 10) ("'163 spec states sequence of components is identified when first packet arrives and disavows prior art 'predefined configuration' (e.g., 1:41–43, 1:64–66, 2:40–45)"). It is appropriate to limit the claims accordingly. *See Inpro II Licensing, S.A.R.L. v. T-Mobile USA Inc.*, 450 F.3d 1350, 1353–57 (Fed. Cir. 2006) (limiting a claim to exclude certain technology when that technology was disparaged in the patent and prosecution history and when overcoming the failing of that technology was described as a purpose of the invention); *see also, Chicago Bd. Options Exch., Inc. v. Int'l Sec. Exch., LLC*, 677 F.3d 1361, 1371–72 (Fed. Cir. 2012) (disparaged technology excluded from claims); *UltimatePointer, L.L.C. v. Nintendo Co.*, 816 F.3d 816, 822–24 (Fed. Cir. 2016) (same).

Claim 9 does not dictate a different result. Rather, this claim should be understood to mean that the sequence of routines may be dynamically configured by selecting preexisting sequences of routines that are combined to form the dynamically generated sequence. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) (en banc) (“The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction.”) (quoting *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998)). The Court addresses Claim 9 in more detail in the section on “the packet of the message” below.

The Court’s understanding of the “sequence of routines” that form the processing sequence comports with the *F5 Networks II* determination that the invention disclosed in the ’683 Patent is limited to sequences that are not configured before receipt of the first packet. The Court need not reach issues relating to estoppel. Further, the Court agrees with the analysis in *F5 Networks II*. Specifically, the statements in the prosecution history of the ’163 Patent are relevant to interpretation of the ’683 and ’790 Patents and that the statements regarding preconfigured sequences of routines are directed to the invention disclosed in the patents, and not specifically to claim language unique to the ’163 Patent. Indeed, the prosecution-history statements explain a disavowal of preconfigured sequences that is stated in the common specification of the ’163, ’683, and ’790 Patents.

Accordingly, the Court construes “sequence of routines” and “sequence of two or more routines” as follows:

- “sequence of routines” means “an ordered arrangement of software routines that was not identified (i.e., configured) prior to receiving a first packet of the message”; and

- “sequence of two or more routines” means “an ordered arrangement of two or more software routines that was not identified (i.e., configured) prior to receiving a first packet of the message.”

A-2. “processing packets” and “process ... packets”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“processing packets” • ’683 Patent Claims 1 • ’790 Patent Claim 8	applying one or more routines to packets	applying one or more conversion routines to packets
“process ... packets” • ’683 Patent Claims 1, 10, 24 • ’790 Patent Claim 8	apply one or more routines to packets	apply one or more conversion routines to packets

Because the parties’ arguments and proposed constructions with respect to these terms are related, the Court addresses the terms together.

The Parties’ Positions

Plaintiff submits packet processing of the claims is not limited to application of *conversion* routines. Dkt. No. 101 at 13–14.

Defendants respond that the Demultiplexing Patents limit packet-processing routines to conversion routines. Dkt. No. 103 at 19–20.

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: ’683 Patent col.2 ll.44–49, col.2 l.65 – col.3 l.1, col.3 ll.17–20, col.4 ll.32–34.

Plaintiff replies that “process” and “processing” are not specially defined as conversion processing and are entitled their full plain and ordinary meaning. Dkt. No. 106 at 8.

Analysis

The parties agree that packet-processing limitations require application of one or more routines to packets. The issue in dispute appears to be whether each routine in the sequence of routines is a conversion routine. While the claims are directed to converting packets from one form to another by processing with the sequence of routines, they do not require that each routine in the sequence is necessarily a conversion routine.

Each claim is explicitly directed to processing data packets to convert from one form to another. For example, '683 Patent Claim 1 recites:

process subsequent packets in the message using the sequence of routines indicated in the stored path, wherein the sequence includes a routine that is used to execute a Transmission Control Protocol (TCP) to convert one or more packets having a TCP format into a different format.

'683 Patent col.14 ll.30–35. Claim 24 recites:

process subsequent packets of the message using the sequence of routines referenced by the one or more data structures, including by removing an outermost header of a given packet using a first routine corresponding to a protocol in a first layer and by removing the resulting outermost header using a second routine corresponding to a different protocol in a different layer.”

Id. at col.16 ll.34–39. Similar conversion-process limitations are found in each independent claim of the '683 and '790 Patents.

Other than Claim 10, however, the claims do not explicitly require that each of the processing routines of the sequence be a “conversion routine.” The Court will not read such a limitation into the claims. That the term “conversion routines” was used in Claim 10—and simply “routines” in the other claims—strongly suggests that the patentee did not intend to limit all the routines to “conversion routines.” *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) (en banc) (noting that the use of the term “steel baffles” “strongly implies that the term ‘baffles’ does not inherently mean objects made of steel”). The Court is not persuaded that the

patents otherwise limited “routines” to conversion routines. Even if all described routines are conversion routines, that alone is not sufficient to inject such a limitation into the claims. *Id.* at 1323 (“we have expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment”); *Thorner v. Sony Comput. Entm’t Am. LLC*, 669 F.3d 1362, 1366 (Fed. Cir. 2012) (“It is likewise not enough that the only embodiments, or all of the embodiments, contain a particular limitation. We do not read limitations from the specification into claims; we do not redefine words. Only the patentee can do that.”); *SRI Int’l v. Matsushita Elec. Corp.*, 775 F.2d 1107, 1121 (Fed. Cir. 1985) (en banc) (“The law does not require the impossible. Hence, it does not require that an applicant describe in his specification every conceivable and possible future embodiment of his invention.”). Finally, Claim 6 recites: “wherein the sequence of routines **includes a routine that is executable to process the packets without converting a format of the packets.**” ’683 Patent col.14 ll.53–55 (emphasis added). Simply, “routine” is broader than “conversion routine.”

Defendants’ argument also seems to exclude processing disclosed in the patents. Specifically, Defendants state that compression is not a conversion routine. Dkt. No. 103 at 20. Yet the patents contemplate that compression (and encryption) routines may be used in the process for transmitting and receiving data in computer-to-computer messaging. ’683 Patent col.1 ll.24–44. That is, the patents disclose that processes other than converting from one transport-layer format to another may be part of the packet processing.

Accordingly, the Court construes “processing packets” and “process … packets” as follows:

- “processing packets” means “applying one or more routines to packets”;
- “process … packets” means “apply one or more routines to packets.”

A-3. “create”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“create” • ’683 Patent Claims 1, 10, 24	plain and ordinary meaning	dynamically generate after the first packet of a message is received

The Parties’ Positions

Plaintiff submits that the “create” of the Demultiplexing Patents may occur at any time—it is not limited either to dynamic generation or creation that occurs only after receipt of the first packet of a message. Dkt. No. 101 at 14.

Defendants respond that the claims themselves require that the “create” happens after receipt of a packet in a message. Dkt. No. 103 at 18–19. Defendants also contend that ’683 Patent distinguishes the prior art on the ground that the invention creates a sequence of routines after receiving the first packet of the message. *Id.* at 19 (citing ’683 Patent col.1 1.45 – col.2 1.11). Finally, Defendants note that their proposed construction is consistent with the disclaimer of preconfigured sequences. *Id.*

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: ’683 Patent col.1 1.45 – col.2 1.11; ’163 Patent File Wrapper September 1, 2009 Amendment and Response to Office Action in Ex Parte Reexamination 90/010,356 (Defendants’ Ex. 1, Dkt. No. 103-1), February 8, 2010 Amendment and Response to Advisory Action in Ex Parte Reexamination 90/010,356 (Defendants’ Ex. 2, Dkt. No. 103-2), October 23, 2009 Interview Summary in Ex Parte Reexamination 90/010,356 (Defendants’ Ex. 4, Dkt. No. 103-4).

Plaintiff replies that the plain and ordinary meaning of “create” does not require “dynamically” creating or creating “after the first packet of a message is received.” Dkt. No. 106

at 8. Rather, Plaintiff contends, Claim 10 expressly recites “obtain information from a *particular* packet,” not necessarily from the first packet. *Id.* (emphasis by Plaintiff).

Analysis

The issue here is the same as the issue addressed in the section on “sequence of routines”; namely, whether the created path/data structure may comprise a preconfigured sequence of routines. It may not.

As explained above, the invention of the ’683 Patent is directed to dynamic configuration of conversion-processing sequences—and does not include preconfiguring the conversion-processing sequence of routines. Thus, the path/data structure comprising the conversion-processing sequence of routines is necessarily created after receipt of the first packet of the message. That said, this limitation is found in the Court’s construction of “sequence of routines” and “list of conversion routines” and does not need to be separately injected into “create.”

Accordingly, the Court determines that “create” has its plain and ordinary meaning without the need for further construction.

A-4. “the packet of the message”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“the packet of the message”	plain and ordinary meaning	indefinite
• ’683 Patent Claim 9		

The Parties’ Positions

Plaintiff submits that “the packet of the message” in Claim 9 of the ’683 Patent refers to “a received packet of a message” in Claim 1, from which Claim 9 ultimately depends. Dkt. No. 101 at 15–16.

Defendants respond that this term lacks antecedent basis. Dkt. No. 103 at 20–22. Specifically, Defendants argue that because Claim 1 recites both “a received packet of a message” and “subsequent packets in the message,” it is unclear which packet of the message is “the packet of the message” in Claim 9. *Id.* at 21–22.

Plaintiff replies that the antecedent basis of “the packet of the message” is Claim 1’s “information in a received packet of a message” because: (1) Claim 8 recites “to identify an address associated with the information,” which refers to the information in the received packet and (2) Claim 9, which depends from Claim 8, recites using this “address” that is associated with “the packet of the message.” Dkt. No. 106 at 9.

Analysis

The issue here is whether the antecedent basis for “the packet of the message” is reasonably certain. In context, it is reasonably certain that antecedent basis is the “received packet of a message” recited in Claim 1.

Claim 9's "the packet of the message" refers to the "received packet of a message" in Claim

1. Claim 1, reproduced here and annotated by the Court, creates and stores a path indicating a sequence of routines "based on information in a received packet of information." As explained above, this sequence of routines is created dynamically on or after receipt of the packet. Claim 1 further processes "subsequent packets in the message" using the created and stored path. Claim 8, which depends from Claim 1, further identifies "an address associated with the information, wherein the address indicates the routines in the sequence of routines of the created path." Thus, (1) the address is associated with the information in

the received packet that triggers creation of the sequence of routines and (2) the address indicates the routines in the created sequence. Claim 9, which depends from Claim 8, recites using this "address to select the sequence of routines from a plurality of sequences of routines that are stored by the first apparatus prior to receiving the packet of the message." Here, "prior to receiving the packet of the message" establishes a time based on receiving the packet. The only reference in the dependency chain to receiving a packet is Claim 1's "received packet of a message." Thus, it is reasonably certain that Claim 9's "the packet of the message" refers to Claim 1's "received packet of a message."

1. A first apparatus for receiving data from a second apparatus, the first apparatus comprising:
a processing unit; and
a memory storing instructions executable by the processing unit to:
create, based on an identification of information in a *received packet of a message*, a path that includes one or more data structures that indicate a sequence of routines for processing packets in the message;
store the created path; and
process subsequent packets in the message using the sequence of routines indicated in the stored path, wherein the sequence includes a routine that is used to execute a Transmission Control Protocol (TCP) to convert one or more packets having a TCP format into a different format.

8. The first apparatus of claim 1, wherein the memory stores instructions executable by the processing unit to identify an address associated with the information, wherein the address indicates the routines in the sequence of routines of the created path.

9. The first apparatus of claim 8, wherein the memory stores instructions executable by the processing unit to use the address to select the sequence of routines from a plurality of sequences of routines that are stored by the first apparatus prior to *receiving the packet of the message*.

This does not mean, however, that Claim 1's "create ... a path" limitation encompasses merely selecting a preconfigured sequence of routines. As stated above, the patents explain that the invention does not include using the prior-art approach of "pre-configured paths, which are defined at 'build-time' before the first packet of a message is received." '163 Patent File Wrapper September 1, 2009 Amendment and Response to Office Action in Ex Parte Reexamination 90/010,356 at 18 (Dkt. No. 103-1 at 18). The Court thus understands that Claim 9 is directed to selecting pre-existing sequences to create the sequence of routines that constitutes the path. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) (en banc) ("The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction.") (quoting *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998))). The Court's understanding comports with the *F5 Network II* explanation that "the invention may rely in some part on 'predefined configuration information'" so long as "it does not identify the complete sequence of ... routines needed to process the packet until *after* a first packet of the message is received." 2015 U.S. Dist. LEXIS 60197, at *40–41.

Accordingly, the Court determines that Defendants have not established that "the packet of the message" renders any claim indefinite.

- "the packet of the message" means "the received packet of the message that was used to create the path."

A-5. “list of conversion routines”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“list of conversion routines” • ’683 Patent Claim 10	An ordered arrangement of software routines for changing data	an ordered arrangement of software routines for converting data from one format to another that was not identified (i.e., configured) prior to receiving a first packet of the message

The Parties’ Positions

Plaintiff submits that for the same reasons “sequence of routines” should not be limited as Defendants propose, “list of conversion routines” should not be limited as Defendants propose. Dkt. No. 101 at 16.

Defendants respond that the disclaimer of preconfigured routines applies to “list of conversion routines.” Dkt. No. 103 at 17. Defendants further respond that a “conversion routine” of the ’683 Patent is not simply a routine for changing data, but rather is a routine for converting data from one format to another format. *Id.* at 17–18.

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: ’683 Patent col.2 ll.42–49, col.2 ll.61–65, col.4 ll.46–48; ’163 Patent File Wrapper September 1, 2009 Amendment and Response to Office Action in Ex Parte Reexamination 90/010,356 (Defendants’ Ex. 1, Dkt. No. 103-1), February 8, 2010 Amendment and Response to Advisory Action in Ex Parte Reexamination 90/010,356 (Defendants’ Ex. 2, Dkt. No. 103-2), October 23, 2009 Interview Summary in Ex Parte Reexamination 90/010,356 (Defendants’ Ex. 4, Dkt. No. 103-4).

Plaintiff replies that there is no disclaimer of preconfigured routines. Dkt. No. 106 at 9. Plaintiff further replies that the ’683 Patent describes “converting” as changing the format and that

Defendants' proposed construction is unhelpful as it simply rewrites "conversion" as "converting."

Id.

Plaintiff cites further **intrinsic evidence** to support its position: '683 Patent col.1 ll.31–44.

Analysis

There are two issues in dispute. First, whether the list of routines is necessarily identified only after receipt of the first packet of the message. Second, whether the conversion routines necessarily convert the format of the data. With respect to the first issue, the Court determines the list of routines is dynamically generated as explained in the section on "sequence of routines" above. With respect to the second issue, the Court determines that a conversion routine changes the form or format of data, not merely the substance of the data.

As explained above, the invention of the '683 Patent is directed to dynamic configuration of conversion-processing sequences—and does not include preconfiguring the conversion-processing sequence of routines. This applies equally to the conversion-processing "list of conversion routines."

A "conversion routine" is one that changes the form of data. As explained in the "Background" section of the patents, data exchange among computers may require the data to take on different forms at different points in the transport route. '683 Patent col.1 l.24 – col.2 l.3. For example, data to be transmitted may first be converted to a compressed-and-encrypted form. *Id.* at col.1 ll.32–35. Then, the compressed-and-encrypted form of the data may be converted into a TCP format and then from a TCP format to an IP format and then to an Ethernet format. *Id.* at col.1 ll.35–39. On receipt, the data may be converted back to the uncompressed-and-unencrypted form and ultimately to the format appropriate for output on the computer. *Id.* at col.1 ll.39–44. While it seems the parties agree the format of the data is changed by a conversion routine, it is unclear

whether Defendants intend “converting data from one format to another” to include changing the form of the data through such as compression or encryption. Dkt. No. 103 at 20 (arguing that compression is not application of a conversion routine). To make it clear that conversion routines are not limited to converting among transport-layer formats and include form-changing processes such as conversion, the Court adopts “changing the form of data.”

Accordingly, the Court construes “list of conversion routines” as follows:

- “list of conversion routines” means “an ordered arrangement of software routines for changing the form of data and that was not identified (i.e., configured) prior to receiving a first packet of the message.”

B. The Applet Patents

B-1. “form of the application”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“form of the application” • ’685 Patent Claims 1, 3, 15, 18, 34, 49	plain and ordinary meaning	source, intermediate or binary format of the application

The Parties’ Positions

Plaintiff submits that “form” is not synonymous with “format” and that the Applet Patents allow that an application may come in a variety of forms. Dkt. No. 101 at 17. Plaintiff argues that the patents allow for forms other than the formats enumerated in Defendants’ proposal. *Id.* For example, Plaintiff contends that Claim 5 refers to “Java byte code” form, Claim 92 refers to “compiled forms,” and Claim 95 refers to “Java source” form. *Id.* at 17–18.

In addition to the claims themselves, Plaintiff cites the following **intrinsic evidence** to support its position: ’685 Patent col.3 ll.44–56.

Defendants respond that the Applet Patents and related prosecution-history statements describe that the form of the application is one of source, intermediate, or binary format. Dkt. No. 103 at 22–24. Defendants argue that the patentee disclaimed any other interpretation of “form of the application” in prosecuting U.S. Patent No. 7,774,740,⁴ which issued from an application in the series of continuation applications connecting the Applet Patents. *Id.* at 22–23. According to Defendants, “Java byte code” is in the intermediate format and the intermediate and binary formats are both compiled forms, thus their proposed construction encompasses the forms recited in the claims. *Id.* at 24.

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: ’685 Patent fig.2A, col.3 ll.12–13, col.3 ll.16–26, col.3 ll.48–51, col.4 l.15, col.4 l.36, col.4 l.60, col.5 ll.40–42, col.6 ll.29–33; U.S. Patent No. 7,774,740 File Wrapper October 14, 2009 Office Action (Defendants’ Ex. 7, Dkt. No. 103-7), December 14, 2009 Response (Defendants’ Ex. 8, Dkt. No. 103-8).

Plaintiff replies that the Applet Patents distinguish “form” from “format” as well as “compiled” from “intermediate” code and disclose that the application may be “any form of program instructions,” so it would be improper to limit the “form of the application” to the three exemplary formats. Dkt. No. 106 at 10–12. Plaintiff further replies that the prosecution-history statements for U.S. Patent No. 7,774,740, when considered in context, are not a disclaimer of any form other than the three exemplary formats and instead disclose a web page as a form of application. *Id.* at 11.

⁴ U.S. Patent No. 7,774,740 is related to the ’685 Patent through a series of continuation applications and the specifications of the two patents are therefore substantially identical except for the claim sets. See ’779 Patent, at [63] Related U.S. Application Data.

Plaintiff cites further **intrinsic evidence** to support its position: '685 Patent col.2 ll.39–45, col.3 ll.10–13; U.S. Patent No. 7,774,740 File Wrapper December 14, 2009 Response (Defendants' Ex. 8, Dkt. No. 103-8).

Analysis

The issue with respect to this term is whether the “application” of the claims of the '685 Patent can take on forms other than source, intermediate, or binary format. As the patent describes, an application is necessarily either uncompiled, partially compiled, or compiled to run on the client computer. These are, respectively, the source, intermediate, and binary formats of the patent. While the application is necessarily in such format, it may take on multiple forms within those formats.

The “form” of an application is not coextensive with the application’s “format.” As explained in the '685 Patent, an applet, and by extension an application,⁵ is “any **form** of program instructions.” '685 Patent col.3 ll.10–15 (emphasis added). These instructions are limited to three **formats**: (1) binary format, compiled for a specific processor, (2) intermediate format, compiled, but not for a specific processor, and (3) source format, uncompiled. *Id.* at col.2 ll.25–44, col.3 ll.9–27, col.3 ll.48–50, col.4 l.66 – col.5 l.14. While limited to three formats, the patent explains that “[t]he form may be selected from a large matrix of many possible forms that can be recognized by the system.” *Id.* at col.46–48. The Court does not understand that three formats constitute a “large matrix.” The form includes format, but also includes factors such as whether it is verified, optimized, or compressed. *Id.* at col.3 ll.44–56. The patents also explain that a request for an applet “is made up of a series of tags which specify the requested applet, the platform on which it is to be run and the type of code (source/intermediate/binary), a verification level and an optimization

⁵ The Court understands that under the parties’ agreed constructions of “application” and “applet,” teachings in the Applet Patents regarding “applets” also apply to “applications.”

level.” *Id.* at col.6 ll.53–56. The Court understands “the platform on which it is to be run” is another exemplary factor of the form of the application. This suggests that the “form” of an applet/application depends on much more than the format (or “type of code” of the applet). Given the number of exemplary form factors, the potential forms of an application may indeed constitute a “large matrix.” Thus, while all application forms are necessarily restricted to three formats (source/intermediate/binary), “form” and “format” are not coextensive.

Further, the patent describes changing the form of an applet without changing the format of the applet. Specifically, the patent describes “transformers” that perform “transformation operations.” *Id.* at col.3 l.64 – col.4 l.1. These transformers “are programs that take in intermediate code and put out intermediate code.” *Id.* at col.5 ll.15–25. The transformers transform (change the form) of the application without changing the format of the application (intermediate code). *Id.* Again, this suggests that form is not defined exclusively by format.

The patentee’s statements in prosecution of a related application do not redefine “form” to be coextensive with “format.” True, the patentee stated: “the Examiner has confused the ‘form’ of the application in the present invention, i.e., whether it is source code, binary code or intermediate code, with the FORM element of HTML discussed in Ferris.” U.S. Patent No. 7,774,740 File Wrapper December 14, 2009 Response at 13 (Dkt. No. 103-8 at 13). In the context of other of the patentee’s prosecution-statements and the shared disclosure of the Applet Patents, equating “form” and “format” would be improper. For example, as discussed above, the ’685 Patent (and U.S. Patent 7,774,740) teach transforming an application without changing its format. *See, e.g.,* ’685 Patent col.5 ll.15–25. The patentee discussed this transformation when prosecuting U.S. Patent No. 7,774,740. *See, e.g.,* December 14, 2009 Response at 15 (Dkt. No. 103-8 at 15). Thus, in the very prosecution response Defendants contend evinces that “form” and “format” are coextensive,

the patentee suggested that “form” and “format” are distinct. To the extent that the prosecution-history sentence Defendants cite equates “form” and “format,” that statement is clearly incorrect and should be disregarded. *See Biotec Biologische Naturverpackungen GmbH v. Biocorp, Inc.*, 249 F.3d 1341, 1348 (Fed. Cir. 2001) (“A person of reasonable intelligence would not be misled into relying on the erroneous statement [in the prosecution history], for it is contrary not only to the plain language of the claims and the specification, but also to other statements in the same prosecution document.”).

Accordingly, the Court determines that the “form of the application” is distinct from the format of the application and “form of the application” has its plain and ordinary meaning without the need for further construction.

B-2. “resource”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“resource”	plain and ordinary meaning	an application or applet
• ’740 Patent Claims 1, 19		

The Parties’ Positions

Plaintiff submits that a “resource” is not limited to “an application or applet.” Dkt. No. 101 at 14. Plaintiff notes that Claim 10 of the ’740 Patent recites that “the resource is a web page” and Claim 11 recites that “the resource is an application.” *Id.* at 19. Plaintiff contends that a “web page” is neither an applet nor an application, so it would be improper to limit “resource” as Defendants propose. Plaintiff also contends that Claim 11 would be rendered superfluous by Defendants’ construction. *Id.*

In addition to the claims themselves, Plaintiff cites the following **extrinsic evidence** to support its position: W3C, *HTML & CSS*, <https://www.w3.org/standards/webdesign/htmlcss>.

Defendants respond that the described invention of the '740 Patent (and all the Applet Patents) is “an applet server computer that fills requests for ‘applets’ and ‘applications’” and “resource” must be construed consistent with what the patents describe as the invention. Dkt. No. 103 at 24–27. Defendants argue that although Claim 10 states that the resource is a web page and a web page is neither an application nor an applet, Claim 10 is not supported by the written description. *Id.* at 16.

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: '685 Patent, at [57] Abstract, col.2 ll.10–14, col.2 l.66 – col.3 l.6, col.3 ll.29–34, col.6 ll.53–56.

Plaintiff replies that the plain meaning of “resource” is well understood, and is not limited to applications and applets. Dkt. No. 106 at 12. Rather, in the context of the Applet Patents, a “resource” is “any data object that may be referred to by a URL.” *Id.*

Plaintiff cites further intrinsic and extrinsic evidence to support its position: **Intrinsic evidence:** '740 Patent figs.2A, 2B, col.6 ll.63–66. **Extrinsic evidence:** Network Working Group, *Request for Comments 2068, Hypertext Transfer Protocol – HTTP/1.1* (Jan. 1997), available at <http://www.ietf.org/rfc/rfc2068.txt> (Plaintiff’s Reply Ex. 2, Dkt. No. 106-2); Network Working Group, *Request for Comments 1738, Uniform Resource Locators (URL)* (Dec. 1994), available at <https://www.ietf.org/rfc/rfc1738.txt> (Plaintiff’s Reply Ex. 3, Dkt. No. 106-3).

Analysis

The issue in dispute is whether “resource” is limited to an application or applet. The Court understands that while a resource necessarily includes an applet or application or code that can be used to build an applet or application, it is not coextensive with an applet or application.

The “resources” described in the Applet Patents “are comprised of program modules 32 (applets in source form, not the requested form) and compilers 30.” ’685 Patent col.4 ll.34–35; ’740 Patent col.4 ll.54–56; *see also*, ’740 Patent at [57] Abstract (“local resources (program code modules and compilers)”). These resources are used to construct applets. ’685 Patent col.3 l.64 – col.4 l.16 (“the applet server manager 22 will attempt to build the requested applet from local resources 26”), col.4 ll.40–41 (“Program modules 32 are program code used to build applets.”);’740 Patent at col.4 ll.17–36, col.4 ll.60–61. The resources may be on the applet server or they may be available on an external network. ’685 Patent at col.4 ll.10–17; ’740 Patent at col.4 ll.29–36. The resources contain the program-module code that can be used to build an applet. ’685 Patent at col.4 ll.41–43 (“Program modules 32 may be stored in local resources 26 in source, binary, or intermediate formats.”); ’740 Patent at col.4 ll.61–62.

The claims of the ’740 Patent are directed to providing applet source-code resources from an external network, not just “any data object that may be referred to by a URL.” The ’740 Patent, like all the Applet Patents, is directed to provision of applets/applications. *See, e.g.*, ’740 Patent col.2 ll.21–25 (“There is a need for a scalable distributed system architecture that provides a mechanism for client computers to request and execute applets in a safe manner without requiring the client machines to have local resources to compile or verify the code.”); *see also, id.* at [57] Abstract (“An applet server accepts requests for applets from client computers.”). The term “resources” must be understood in this context. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) (*en banc*) (“The claims are directed to the invention that is described in the specification; they do not have meaning removed from the context from which they arose.” (quoting *Netword, LLC v. Centraal Corp.*, 242 F.3d 1347, 1352 (Fed. Cir. 2001))). Each claim of the ’740 Patent recites “a resource, wherein the resource includes source code.” ’740 Patent col.7 ll.27–28 (Claim

1), col.8 ll.60–61 (Claim 19), col.9 ll.7–8 (Claim 20). Each claim further recites “conveying ... a request for the resource to an external network.” *Id.* at col.7 ll.31–32 (Claim 1), col.8 ll.64–65 (Claim 19), col.9 ll.11–12 (Claim 20). This claim language closely parallels the described embodiments in which the server collects “local resources” from an external server to use those resources to provide the application that client requests. Thus, while the resources may take on various forms, such as a web page (Claim 10) or an application (Claim 11), the resources all contain source code that comprises an application or applet or can be used to build an application or applet.

Accordingly, the Court construes “resource” as follows:

- “resource” means “data object containing code that: (1) is an application, (2) is an applet, or (3) can be used to build an application or applet.”

B-3. “generating the identified form of the application from another form of the application” and “generated the identified form of the application from another form of the application”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“generating the identified form of the application from another form of the application” • ’685 Patent Claims 1, 3, 15, 18	plain and ordinary meaning	building or producing the form of the application from local resources on the server through the compilation of one or more program code modules using one or more compilers, and optionally applying one or more transformation operations
“generated the identified form of the application from another form of the application” • ’685 Patent Claims 34, 49		

Because the parties’ arguments and proposed constructions with respect to these terms are related, the Court addresses the terms together.

The Parties' Positions

Plaintiff submits that the claims separately recite the “how and why” of “generating the identified form” and such details should not be read into the construction of “generating the identified form of the application from another form of the application” and “generated the identified form of the application from another form of the application.” Dkt. No. 101 at 20.

Defendants respond that its proposed construction is the same construction that Plaintiff proposed in a previous litigation and is as described in the Applet Patents. Dkt. No. 103 at 27–28 (citing Joint Claim Construction and Prehearing Statement, *Implicit Networks, Inc. v. Sybase, Inc. and Microsoft Corporation*, No. 3:09-cv-1478-SI (N.D. Cal. Jan. 11, 2010), Dkt. No. 46 (Dkt. No. 103-9)). Specifically, Defendants contend that the patents describe generating the form “necessarily involves the compilation of one or more program code modules using one or more compilers.” *Id.* at 28.

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: ’685 Patent col.3 ll.16–20, col.3 ll.24–27, col.3 l.57 – col.4 l.9, col.4 ll.56–65.

Plaintiff replies that Defendants’ reliance on the Joint Claim Construction and Prehearing Statement in the Microsoft litigation is misplaced because: (1) judicial estoppel does not apply because Plaintiff’s positions were not adopted, and (2) the claims at issue here are distinct from those at issue in Microsoft in that Claim 3 requires “requesting the application in the other form from a computer other than the server computer” which contradicts the notion that the application must be generated on the server. Dkt. No. 106 at 12–13.

Analysis

The issue in dispute here is whether generating the identified form of the application from another form of the application necessarily entails building the form of the application from resources on the server through compilation of program modules. It does not.

The '685 Patent discloses multiple ways in which the server can provide the requested application in the form requested. As explained in the Applet Patents, “[t]he request [for an applet] is made up of a series of tags which specify the requested applet, the platform on which it is to be run and the type of code (source/intermediate/binary), a verification level and an optimization level.” '685 Patent col.6 ll.53–56. The patent also provides compression as another exemplary form factor. *Id.* at col.3 ll.50–53. The combination of such form factors specifies the “form” of the applet (or application). On request for an applet/application, the server can identify an existing application in its cache. *Id.* at col.3 l.57 – col.4 l.16. If no suitable applet/application exists, the server can build one from local resources and transformers on the server. *Id.* It may also request the applet/application, or the required resources and transformers, from an external network. *Id.* Transformers are used to change the form of the applet/application but are distinct from compilers—transformers change the form without changing the format as would the compiler. *Id.* at col.4 l.66 – col.5 l.25 (explaining compilers and transformers); *see also, id.* at col.5 l.26–53 (verifier transformer), col.5 l.66 – col.6 l.6 (optimizer transformer). Thus, the patent contemplates generating the requested form of an application, e.g., a verified application, by acquiring the application from an external network and running it through a transformer, e.g., a verifier transformer. Defendants’ proposed construction does not encompass this generation of the requested form of the application.

Accordingly, the Court rejects Defendants' proposed construction and determines that "generating the identified form of the application from another form of the application" and "generated the identified form of the application from another form of the application" each has its plain and ordinary meaning without the need for further construction.

B-4. "source code that is in a form based on the specified one or more client parameters for the first client computer"

Disputed Term	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"source code that is in a form based on the specified one or more client parameters for the first client computer" • '779 Patent Claim 1, 18	plain and ordinary meaning	source code that has been optimized, verified and/or compressed based on the one or more client parameters specified in the received applet request

The Parties' Positions

Plaintiff submits there is no support for limiting the form of the source code to one "that has been optimized, verified and/or compressed." Dkt. No. 101 at 20–21.

Defendants respond that the only relevant client parameters described in the Applet Patents refer to the transformation operations, and the only transformation operations disclosed are optimization, verification, and compression. Dkt. No. 103 at 31–32.

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: '685 Patent col.6 ll.7–12.

Plaintiff replies that claims do not recite variants of "optimize," "verify," or "compressed" and it would be improper to limit the claims to require such. Dkt. No. 106 at 13.

Analysis

The issue here is whether the form of source code based on client parameters is limited to optimized, verified, or compressed. It is not.

While the Applet Patents describe that the client may specify optimization, verification, or compression, the patents do not limit client parameters to those three. As explained in the patents, “[t]he request [for an applet] is made up of a series of tags which specify the requested applet, the platform on which it is to be run and the type of code (source/intermediate/binary), a verification level and an optimization level.” ’685 Patent col.6 ll.53–56; ’779 Patent col.7 ll.18–21. The patents also provide compression as another exemplary form factor. ’685 Patent col.3 ll.50–53; ’779 Patent col.4 ll.19–23. Notably absent from Defendants’ list of limitations is the exemplary “platform on which it is to be run.” This is important to understand Claim 3 of the ’779 Patent, which recites “wherein the form of the source code is suitable for running on the first client computer.” ’779 Patent col.7 ll.56–57. Further, the patents provide that “[v]erification, optimization and compression are *examples* of types of transformation operations.” ’685 Patent col.3 ll.52–53; ’779 Patent col.4 ll.21–23. Even without mention of the “platform on which it is to be run,” it would be improper to limit the claims to transformation operations, and their associated client parameters, to items expressly labeled as “examples.” *See Thorner v. Sony Computer Entm’t Am., LLC*, 669 F.3d 1362, 1367 (Fed. Cir. 2012) (“The patentee is free to choose a broad term and expect to obtain the full scope of its plain and ordinary meaning unless the patentee explicitly redefines the term or disavows its full scope.”). Simply, the Applet Patents do not limit the list of client parameters as Defendants contend.

Accordingly, the Court rejects Defendants’ proposed construction and determines that the term has its plain and ordinary meaning without the need for further construction.

B-5. “a specific form of the particular applet that includes source code, based on the specified one or more parameters in the applet request, wherein the specific form complies with the specified one or more parameters”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“a specific form of the particular applet that includes source code, based on the specified one or more parameters in the applet request, wherein the specific form complies with the specified one or more parameters” • ’779 Patent Claim 12	plain and ordinary meaning	applet that includes source code that has been optimized, verified and/or compressed based on the one or more client parameters specified in the received applet request

The Parties’ Positions

Plaintiff submits there is no support for limiting the form of the applet’s source code to one “that has been optimized, verified and/or compressed.” Dkt. No. 101 at 22.

Defendants respond that the only relevant client parameters described in the Applet Patents refer to the transformation operations, and the only transformation operations disclosed are optimization, verification, and compression. Dkt. No. 103 at 32–33.

Plaintiff replies that claims do not recite variants of “optimize,” “verify,” or “compressed” and it would be improper to limit the claims to require such. Dkt. No. 106 at 13–14.

Analysis

The issue in dispute here is the same as addressed in the section on “source code that is in a form based on the specified one or more client parameters for the first client computer.” For the same reasons given in that section, the Court rejects Defendants’ proposed construction and determines that the term has its plain and ordinary meaning without the need for further construction.

B-6. “transformation operation”

Disputed Term	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“transformation operation” <ul style="list-style-type: none">• ’779 Patent Claim 16• ’740 Patent Claims 1, 19	an operation that analyzes and/or modifies code, including for example through verification, optimization, or compression	an operation that verifies, optimizes or compresses code within an application or applet

The Parties’ Positions

Plaintiff submits the “transformation operation” of the Applet Patents encompasses more than verification, optimization, or compression. Dkt. No. 101 at 23. Plaintiff argues that the patents expressly state that these three transformations are exemplary: “[v]erification, optimization, and compression are examples of types of transformation operations.” *Id.* (quoting ’779 Patent col.4 ll.19–25).

In addition to the claims themselves, Plaintiff cites the following **intrinsic evidence** to support its position: ’779 Patent col.4 ll.19–25, col.5 ll.51–61.

Defendants respond that all transformation operations described in the Applet Patents operate on code. Dkt. No. 103 at 29–30. Defendants contend that the described transformations are limited to verification, optimization, and compression that involves code-level evaluation. *Id.* at 30–31.

In addition to the claims themselves, Defendants cite the following **intrinsic evidence** to support their position: ’685 Patent col.5 ll.15–20; ’779 Patent col.4 ll.19–25, col.5 ll.51–61.

Plaintiff replies that the “transformation operation” of Claims 1 and 19 of the ’740 Patent applies to a “resource” rather than “an application or applet” so it would be improper to limit the transformation operation to applications or applets. Dkt. No. 106 at 14.

Analysis

There are two issues in dispute. First, whether a “transformation operation” is necessarily limited to verification, optimization, or compression. Second, whether the “transformation operation” is necessarily limited to applets and applications. With respect to the first issue, transformation operations are not so limited. With respect to the second issue, transformation operations may operate on any code, whether the code is stand-alone, associated with a larger application, or otherwise.

As explained above in the section on “source code that is in a form based on the specified one or more client parameters for the first client computer,” verification, optimization, and compression are provided as exemplary transformation operations, not as an exhaustive list of operations. ’685 Patent col.3 ll.52–53 (“Verification, optimization and compression are examples of types of transformation operations.”) The Court will not limit transformation operations to this list of “examples.”

The Applet Patents teach that transformation operations may be applied to applets and applications as well as to program modules that are used to build applets. ’685 Patent col.3 l.44 – col.4 l.16. The program modules are code fragments that may be combined to create an applet. *See, e.g., id.* at col.4 ll.3–6 (“The requested applet is built by selecting one or more program code modules 32 and compiling them with one or more compilers 30.”), col.4 ll.40–41 (“Program modules 32 are program code used to build applets.”). These modules may be “transformed” to put in a suitable form. *See, e.g., id.* at col.4 ll.56–65. Thus, transformation operations may be performed on code that can be used to build applets or applications, and not merely on applets or applications themselves.

Transformation operations require modification of code. The Applet Patents describe a verifier transformer “that is able to analyze input code and determine areas that might not be safe.” *Id.* at col.5 ll.26–27. While the transformer may be capable of performing an “analyze” operation without transforming code, that capability does not define the verifier’s transformation operation. Rather, the transformation operation performed by the verifier occurs once it “determines the portion of the unsafe applet code.” *Id.* at col.5 ll.31–40. For example, the verifier may modify the code such that the “offending portion of the code can be encased with new code that specifically prevents this unsafe code from being executed.” *Id.* at col.5 ll.32–35. Or the code may be modified such that “unsafe code [is] flagged in such a way that a user can be warned about the possible risks of executing the code fragment.” *Id.* at col.5 ll.35–38. Ultimately, “[v]erifiers 48 are responsible for identifying non-secure portions of code in the intermediate code and modifying this code to make it secure.” *Id.* at col.5 ll.48–50. That is, the transformation operation of the verifier, like all transformation operations, transforms code by modifying the code.

Accordingly, the Court is not persuaded that the term should be limited as Defendants contend and construes “transformation operation” as follows:

- “transformation operation” means “operation that modifies code.”

V. CONCLUSION

The Court adopts the constructions set forth above, as summarized in the following table. The parties are **ORDERED** that they may not refer, directly or indirectly, to each other’s claim construction positions in the presence of the jury. Likewise, the parties are **ORDERED** to refrain from mentioning any portion of this opinion, other than the actual definitions adopted by the Court, in the presence of the jury. Any reference to claim construction proceedings is limited to informing the jury of the definitions adopted by the Court.

So Ordered this

Mar 29, 2017



RODNEY GILSTRAP
UNITED STATES DISTRICT JUDGE